Name: Santos, Tristan Neal U.
Course and Section: CPE32S9
Date of Submission: 02-21-2024
Instructor: Engr. Roman Richard

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
#import pandas
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
#create a pandas dataframe called "training" from the titanic-train.csv file
training = pd.read_csv("/content/drive/MyDrive/Colab Notebooks/Emtech2/titanic_train.csv")
test = pd.read_csv("/content/drive/MyDrive/Colab Notebooks/Emtech2/titanic_test.csv")
```

```
 #Code cell 2
#verify the contents of the training dataframe using the pandas info() method.
#training.?
training.describe()
```

| | PassengerId | Survived | Pclass | Age | SibSp | Parch | Fare |
|---|---|---|---|---|---|---|---|
| **count** | 891.000000 | 891.000000 | 891.000000 | 714.000000 | 891.000000 | 891.000000 | 891.000000 |
| **mean** | 446.000000 | 0.383838 | 2.308642 | 29.699118 | 0.523008 | 0.381594 | 32.204208 |
| **std** | 257.353842 | 0.486592 | 0.836071 | 14.526497 | 1.102743 | 0.806057 | 49.693429 |
| **min** | 1.000000 | 0.000000 | 1.000000 | 0.420000 | 0.000000 | 0.000000 | 0.000000 |
| **25%** | 223.500000 | 0.000000 | 2.000000 | 20.125000 | 0.000000 | 0.000000 | 7.910400 |
| **50%** | 446.000000 | 0.000000 | 3.000000 | 28.000000 | 0.000000 | 0.000000 | 14.454200 |
| **75%** | 668.500000 | 1.000000 | 3.000000 | 38.000000 | 1.000000 | 0.000000 | 31.000000 |
| **max** | 891.000000 | 1.000000 | 3.000000 | 80.000000 | 8.000000 | 6.000000 | 512.329200 |

```
test.describe()
```

| | PassengerId | Pclass | Age | SibSp | Parch | Fare |
|---|---|---|---|---|---|---|
| **count** | 418.000000 | 418.000000 | 332.000000 | 418.000000 | 418.000000 | 417.000000 |
| **mean** | 1100.500000 | 2.265550 | 30.272590 | 0.447368 | 0.392344 | 35.627188 |
| **std** | 120.810458 | 0.841838 | 14.181209 | 0.896760 | 0.981429 | 55.907576 |
| **min** | 892.000000 | 1.000000 | 0.170000 | 0.000000 | 0.000000 | 0.000000 |
| **25%** | 996.250000 | 1.000000 | 21.000000 | 0.000000 | 0.000000 | 7.895800 |
| **50%** | 1100.500000 | 3.000000 | 27.000000 | 0.000000 | 0.000000 | 14.454200 |
| **75%** | 1204.750000 | 3.000000 | 39.000000 | 1.000000 | 0.000000 | 31.500000 |
| **max** | 1309.000000 | 3.000000 | 76.000000 | 8.000000 | 9.000000 | 512.329200 |

```
#Code cell 3
#view the first few rows of the data
training.head()
```

| | PassengerId | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket | Fare | Cabin | Embarked |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 0 | 3 | Braund, Mr. Owen Harris | male | 22.0 | 1 | 0 | A/5 21171 | 7.2500 | NaN | S |
| **1** | 2 | 1 | 1 | Cumings, Mrs. John Bradley (Florence Briggs Th... | female | 38.0 | 1 | 0 | PC 17599 | 71.2833 | C85 | C |
| **2** | 3 | 1 | 3 | Heikkinen, Miss. Laina | female | 26.0 | 0 | 0 | STON/O2. 3101282 | 7.9250 | NaN | S |

Next steps: View recommended plots

```
#code cell 4
training["Sex"] = training["Sex"].apply(lambda toLabel: 0 if toLabel == 'male' else 1)
```

```
 #code cell 5
#view the first few rows of the data again
training.head()
```

| | PassengerId | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket | Fare | Cabin | Embarked |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 0 | 3 | Braund, Mr. Owen Harris | 0 | 22.0 | 1 | 0 | A/5 21171 | 7.2500 | NaN | S |
| **1** | 2 | 1 | 1 | Cumings, Mrs. John Bradley (Florence Briggs Th... | 1 | 38.0 | 1 | 0 | PC 17599 | 71.2833 | C85 | C |
| **2** | 3 | 1 | 3 | Heikkinen, Miss. Laina | 1 | 26.0 | 0 | 0 | STON/O2. 3101282 | 7.9250 | NaN | S |

Next steps: View recommended plots

```
test.head()
```

| | PassengerId | Pclass | Name | Sex | Age | SibSp | Parch | Ticket | Fare | Cabin | Embarked |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 892 | 3 | Kelly, Mr. James | male | 34.5 | 0 | 0 | 330911 | 7.8292 | NaN | Q |
| **1** | 893 | 3 | Wilkes, Mrs. James (Ellen Needs) | female | 47.0 | 1 | 0 | 363272 | 7.0000 | NaN | S |
| **2** | 894 | 2 | Myles, Mr. Thomas Francis | male | 62.0 | 0 | 0 | 240276 | 9.6875 | NaN | Q |
| **3** | 895 | 3 | Wirz, Mr. Albert | male | 27.0 | 0 | 0 | 315154 | 8.6625 | NaN | S |
| **4** | 896 | 3 | Hirvonen, Mrs. Alexander (Helga E Lindqvist) | female | 22.0 | 1 | 1 | 3101298 | 12.2875 | NaN | S |

Next steps: View recommended plots

```
 #code cell 6
training["Age"].fillna(training["Age"].mean(), inplace=True)



#code cell 7
#verify that the missing values for the age variable have been eliminated.
training.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    int64
 5   Age          891 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(6), object(4)
memory usage: 83.7+ KB
```

```
test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  418 non-null    int64
 1   Pclass       418 non-null    int64
 2   Name         418 non-null    object
 3   Sex          418 non-null    object
 4   Age          332 non-null    float64
 5   SibSp        418 non-null    int64
 6   Parch        418 non-null    int64
 7   Ticket       418 non-null    object
 8   Fare         417 non-null    float64
```

```
 9   Cabin        91 non-null     object
 10  Embarked     418 non-null    object
dtypes: float64(2), int64(4), object(5)
memory usage: 36.0+ KB
```

```
 #code cell 8
#create the array for the target values
y_target = training["Survived"].values
```

```
#code cell 9
columns = ["Fare", "Pclass", "Sex", "Age", "SibSp"]
#create the variable to hold the features that the classifier will use
X_input = training[list(columns)].values
```

```
#code cell 10
#import the tree module from the sklearn library
from sklearn import tree

#create clf_train as a decision tree classifier object
clf_train = tree.DecisionTreeClassifier(criterion="entropy", max_depth=3)

#train the model using the fit() method of the decision tree object.
#Supply the method with the input variable X_input and the target variable y_target
clf_train = clf_train.fit(X_input, y_target)
```

```
#code cell 11
clf_train.score(X_input,y_target)
```

```
0.8226711560044894
```

```
#code cell 12
from six import StringIO
with open("/content/drive/MyDrive/Colab Notebooks/Emtech2/titanic_test.csv, 'w') as f:
testing = tree.export_graphviz(clf_train, out_file=f, feature_names=columns)
```

```
  File "<ipython-input-44-5675563abf1c>", line 3
    with open("/content/drive/MyDrive/Colab Notebooks/Emtech2/titanic_test.csv, 'w') as f:
                ^
SyntaxError: unterminated string literal (detected at line 3)
```

```
training.describe()
```

| | PassengerId | Survived | Pclass | Sex | Age | SibSp | Parch | Fare |
|---|---|---|---|---|---|---|---|---|
| **count** | 891.000000 | 891.000000 | 891.000000 | 891.000000 | 891.000000 | 891.000000 | 891.000000 | 891.000000 |

```
test.describe()
```

| | PassengerId | Pclass | Age | SibSp | Parch | Fare |
|---|---|---|---|---|---|---|
| **count** | 418.000000 | 418.000000 | 332.000000 | 418.000000 | 418.000000 | 417.000000 |
| **mean** | 1100.500000 | 2.265550 | 30.272590 | 0.447368 | 0.392344 | 35.627188 |
| **std** | 120.810458 | 0.841838 | 14.181209 | 0.896760 | 0.981429 | 55.907576 |
| **min** | 892.000000 | 1.000000 | 0.170000 | 0.000000 | 0.000000 | 0.000000 |
| **25%** | 996.250000 | 1.000000 | 21.000000 | 0.000000 | 0.000000 | 7.895800 |
| **50%** | 1100.500000 | 3.000000 | 27.000000 | 0.000000 | 0.000000 | 14.454200 |
| **75%** | 1204.750000 | 3.000000 | 39.000000 | 1.000000 | 0.000000 | 31.500000 |
| **max** | 1309.000000 | 3.000000 | 76.000000 | 8.000000 | 9.000000 | 512.329200 |

```
Pass = testFeame[(testFrame.Sex == 'Male')]
```